CSUS EEE 193B / CPE 191 Lab - Section 3

Precision Robotic Manipulator Controller

Deployable Prototype Documentation

Authors: Charles Relat, Amy Giovinazzo, Jamesson Kaupanger and Joseph Smith

SACRAMENTO STATE

4/30/2017

Deployable Prototype Documentation

Amy Giovinazzo (amcferran4@gmail.com), Joseph Smith (josephsmith@ieee.org),
Charles Relat (charles.relat@gmail.com), and Jamesson Kaupanger (jkaupanger7@comcast.net)

Contents

List of Tables

List of Figures

Abstract—
New and evolving technology in the field of robotics requires innovative controllers to control the complex manipulators currently under development using intuitive user input. One way to design a controller with intuitive user input is to use absolute orientation sensors on a user's arm to track hand movement and move a robotic arm proportionally. This paper looks into controlling the Titan 4 robotic manipulator using a controller based on absolute orientation Inertial Measurement Units.

Keywords—
TechnipFMC, Titan 4 controller, Edison board, Remotely Operated Vehicle (ROV), BNO055, UDP communication, and simulator

I. Introduction

This senior design project is fully sponsored by TechnipFMC, formally known as FMC Technologies Schilling Robotics at the start of the project. The focus of this project is designing and implementing a robotic arm controller that is based on human arm movements. This is a group project with contributions from all team members resulting in a final product that solves a societal problem and meets or exceeds all initial design criteria. Many hours are put into solving complex problems through testing, measurement and calculations to arrive at a final solution.

II. Societal Problem

The societal problem at the focus of this project is the need for precision control, specifically in under water robot applications. An underwater vehicle produced by TechnipFMC is shown in Figure 1 with the manipulators that need a precision controller on the front. They have a 6 foot radius for their workspace and 6 degrees of freedom (DOF) so they can reach many positions with different orientations of their end effectors.

Fig. 1. TechnipFMC underwater robot

With TechnipFMC's current controller being a "master arm" design as seen in Figure 2, it takes an operator at least six months or more of training to safely control the robotic arm. This proves to be expensive and time-consuming for the companies who purchase the Titan 4 hydraulic arm. With a new controller that is based on human arm movements, it will be more intuitive and more efficient to use, thus decreasing training time and save money. This new controller will also be equal to the master arm controller in precision.

Fig. 2. TechnipFMC current controller

There is a need for the development of a new controller that will be easily adaptable to new designs of robotic manipulators and different arrangements of joints for underwater exploration, research, and maintenance.

III. Design Idea Contract

By establishing a set of design requirements and constraints, the team is able to focus the design process on the high risk and important features necessary to satisfy the expectations of our sponsor, TechnipFMC. This also allows for the creation of a timeline to ensure the project stays on track and can be completed on time.

By considering a range of design ideas, the most appropriate, achievable, and best fitting design is chosen for implementation to satisfy the design requirements. The resulting design idea consists of a three-part design that uses a main controller in the form of 3 sensors connected to different parts of the user's arm as shown in Figure 3, an enclosure that sits on the ground with a microcontroller and breakout board for all of the sensor connections, and a remote unit that is held in either hand. The Intel Edison microcontroller then sends the movement information through User Datagram Protocol (UDP) packets to the Titan 4 arm.

1. Main Controller

Inertial Measurement Units (IMUs) are used for the main controller to determine hand and arm positions and map the user's hand coordinates in space to the Titan 4 robotic arm's location in space. This is accomplished using two IMUs, one for each part of the arm that is located at positions 1 and 2 in Figure 3, and associating a vector to each section of arm. The vectors are then summed for a final position in space which is the target location for the end effector of the Titan 4 arm.

A third IMU is attached to the back of the user's hand in position 3 of Figure 3 and determines the orientation of the end effector. The IMUs already give angular data so no math is required to convert hand orientation to end effector orientation. [1]

The opening, closing, and rotation of the end effector is controlled using a joystick on the secondary module: up

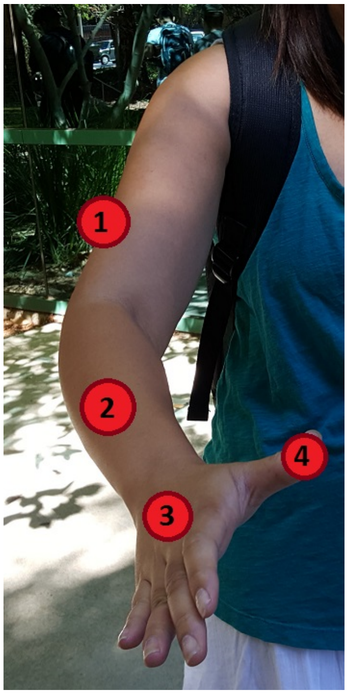

Fig. 3. Approximate placement for each of the sensors being used.

and down control opening and closing, and left and right control rotation. Initially a proximity sensor was to be used for opening and closing, but the necessary precision was not attainable and as a result, the control of the Titan 4 jaws are now done by the secondary module.

The main controller, comprising of the three IMUs, is connected to the hand and arm by means of elastic and Velcro. The wire connecting the main controller to the microcontroller enclosure is a standard category 6 cable with RJ-45 connectors on each end for easy removal.

2. **Microcontroller Enclosure**

The Intel Edison board collects and interprets data from the IMU sensors that are placed on the user's arm. This Edison microcontroller is connected to the IMU sensors through printed circuit boards (PCBs) that communicate through an inter-integrated circuit (I2C) bus. These items are in turn housed inside a 3D printed enclosure. There is a breakout board that connects with the remote unit and main controller using RJ-45 connectors and Ethernet cable, which allows for quick disconnection at the user's arm and easy replacement of a damaged cable.

To communicate with the Titan 4 arm, UDP packets are sent over Ethernet that contain serial data for control of the arm. This is accomplished using the built-in WiFi on the Edison board. [2]

3. **Remote Unit**

The primary function of the remote unit is to allow a lockout function so the user can activate and deactivate movement without having to move the main controller. Both of the types of movement - the end effector opening or closing and movement in the XYZ field - can be locked independently of each other using the lockout button on the remote unit. When a locked movement is unlocked by pressing the associated button, the current position of the associated sensors becomes referenced to the point where it was locked. This means that the user's reference point has changed and the Titan 4 arm will not move until the sensors detect movement from the new reference position.

In addition to the lockout button, the remote unit has a second button that changes the scaling between user arm movement and robotic arm movement. There are three modes which allow for a 1:1, 2:1, and 4:1 movement ratio. Repeated presses of the same button cycles through the three modes available. LED lights convey current settings to the user. In addition, an excel document is provided to map the color correlations for quick referencing.

The primary reason for using a remote unit for button control is to prevent the need to press a button on the control arm that would cause unwanted movement before the lockout occurs. The LEDs are on the device enclosure primarily because users could have their main controller arm twisted in a way that does not allow for them to see the LEDs without unwanted movement but the device enclosure can be place in direct line of sight.

The following resources are required to implement the design idea.

- A dedicated lab space for testing, building and implementing the controller prototype. Additionally, storage space in which to hold the parts that are required to build the controller.
    - Intel Edison micro controller (EDI2ARDUIN.AL.K)
    - 9DOF IMU BNO055 with 9 degrees of freedom Inertial Measurement Unit.
    - Nintendo Nunchuck device. (B0094X2066)
    - LED RGB Strip - WS2812 Serial RGB LED strip - 1 meter.
- Software used: Linux environment for C programming language and debugging and proprietary software provided by TechnipFMC to simulate the Titan 4 robotic arm for testing.

Special Features

1. **Standard Movement**

The standard movement is the default feature for the Titan 4 Controller. This feature performs standard movement of the Titan 4 robotic arm.
Three IMUs are connected with 4-pin wires and elastic material and Velcro are used to hold the sensors in place. An Edison micro-controller (EDI2ARDUIN.AL.K) is needed for communications and calculations.
Software used for this feature includes the Linux environment for coding and debugging and proprietary software provided by TechnipFMC for testing.
Charles implemented vector positioning using the Accelerometer and Magnetometer and end effector orientation using the Gyroscope, Joseph implemented the UDP packet transmissions, and Jamesson completed on the gripping function of the end effector linked to the remote unit with assistance from the team.

2. **Precision Control**

Precision control is an advanced feature for the Titan 4 controller. This feature allows precision movement of the Titan 4 robotic arm. Precision control results in minute movement of the Titan 4 robotic arm relative to standard user arm movement in varying rations of 1:1, 2:1, and 4:1.
The hardware required is the same needed for standard movement with the addition of a secondary module to enable or disable this feature.
Software used for this feature includes the Linux environment for programming and debugging and proprietary software provided by TechnipFMC for testing the features implementation.
Primary work for this feature consists of the reduced movement based on angular movement and the secondary module control. Because of this, precision movement is mostly the work of Charles, however, the team did assist in some instances. In addition, Jamesson is responsible for the secondary module construction, which controls the precision settings.

3. **Position Reset**

Initially it was planned that a button would control the position reset for the arm, however, after testing it was decided that simply locking out the arm movement would reset the position automatically. This is done to add to the ease of use of the device. This feature automatically resets the operators arm when coming back from lockout without adjusting the position of the Titan 4.
Software used for this feature includes the Linux environment for programming and debugging and proprietary software provided by FMC for testing the features implementation.
Charles and Jamesson are responsible for this feature's implementation as it relates to IMU position and secondary module manipulation.

4. **Lockout**

This feature reads a button press to stop or resume arm movement. To ensure the communication connection is not lost, data transmission of the Titan 4 loops with the most recent position information.
This button is located on the secondary module and pressing it changes the lockout state.
Software required for this feature includes the Linux environment for coding and debugging and proprietary software provided by TechnipFMC for testing.
Jamesson and Amy are the primary members responsible for this feature's implementation.

5. **UDP data transfer**

This feature reads the data from the Titan 4 controller and converts it into UDP packets to be sent to the Titan 4 simulator.
An Edison board with Wi-Fi communication is necessary to implement this feature.
Software required for this feature includes the Linux environment for coding and debugging and proprietary software provided by TechnipFMC for testing.
Joseph is the primary group member responsible for this feature as he is responsible for the UDP communications section.

6. **System and Parts Controls**

This feature operates and manipulates the Titan 4 controller itself. For example, the buttons involved to select different features, or hardware necessary for transmission of data.
This is the physical building of the three main parts of the project, so all hardware listed from the previous features is required.
No software components are required to implement this feature.
The entire team is responsible for implementing this feature, differing components were addressed as needed by team members to complete the construction of the controller.

7. **Ease of Use**

During the length of the design process and construction, the ability to easily work with and control the Titan 4 arm is the main focus. By creating an intuitive design and interface, the controller is able to be used seamlessly by operators to maneuver the Titan 4 robotic arm.

Fall 2016 and Spring 2017 Design Goals

In the first semester of senior design, milestones are set to ensure progress is made and the rapid prototype is delivered on time. First, it is required that data be collected from the user's arm movement, and it needs to be made up of enough points of data to convert that information into movement for the Titan 4 arm. Second, the data is interpreted and converted to usable data for the Titan 4 arm, and it is transmitted to the arm by way of UDP packets. Thirdly, the controller needs a way to allow the user to move their arm without making adjustments to the manipulator, so a secondary module is implemented to allow lockout features.

The above referenced design ideas and goals for the Fall 2016 senior design class are satisfied by the end of the Fall 2016 semester.

The Spring 2017 senior design semester is for refining the working prototype into a more polished product. Because nearly all of the functionality of the Titan 4 controller is completed as part of the rapid prototype, the plan for Spring semester is to make improvements to the current design. However, compatibility issues between the Intel Edison and the BNO055 sensors require many hours of testing and re-design that are not initially anticipated. Code for communication between the BNO055 and Edison is now all custom written by the group for this project from scratch. In addition to this, after successful communications between the Edison and the BNO055 sensors is established, the reliability of the data is not acceptable. Many hours are required to test new circuit designs and finally an acceptable interconnection circuit is found.

With the final design, the prototype is using the Intel Edison micro-controller for all of its communications rather than the rapid prototype design that included a Propeller and Arduino board. With this, refinements are implemented into the programming aspects of the device so that the coding reflects a more professional aspect.

The Titan 4 controller is also transformed into a more refined image. An 3D printed enclosure now houses the micro-controller. The secondary module is made from a modified Nintendo Nunchuck remote. Ethernet cables replace loose 8 wire bundles for the controller and remote unit and the wires between the sensors on the main controller are all inside nylon braided protection sleeves. These changes create a polished look for the final product compared to the previous rapid prototype design.

As a final goal for the Spring semester, refinements are planned for the design of the robotic controller to a degree of professionalism expected by TechnipFMC. These plans are successfully implemented, allowing the controller to operate physical Titan 4 manipulators.

IV. Funding

This project is fully funded by TechnipFMC so all parts in tables 1 and 2 were purchased by them and picked up from their office. No parts were purchased through other means. A total of 1,193.82 dollars has been spent in the design and testing process. Due to time constraints and knowing the maximum budget, extra of each part was purchased to avoid delays caused by malfunctioning parts. This strategy also allows multiple team members to work on separate features that require the same hardware simultaneously, cutting down development and testing time. Some parts are also not included in the final design but they are vital to the rapid prototyping portion of the project. As listed at the bottom of Table 1, the total cost of parts before tax and shipping comes to 1,193.82 dollars.
The specific list of parts in the final prototype is listed in Table 4 which is under the Hardware section of Appendix B. The total spent on the parts in the final prototype is 289.02 dollars.

The only changes from the original budget were the additions of a couple of new parts. For instance, the original 9 DOF IMU proved to problematic to use to detect hand motion and orientation. A new part, the BNO055, was found after research, and three were purchased.

Code already existed for the BNO055 sensor for the Arduino micro-controller, so to expedite the process of constructing a working prototype; an Arduino UNO was purchased and implemented in the design of the rapid arm prototype.

V. Project milestones

There are main milestones for both the Fall semester and the Spring semester that needed be fulfilled to call this project successful. These milestones include:

Fall semester:

- Successfully integrate position, orientation, and grasping action of the end effector. Since each of these factors requires their own sensor, it takes great effort to combine these sensor's data together into a readable string of output data that can be sent to the simulator through UDP packets. One BNO055 IMU is used for the orientation, two LSM9DS1 IMUs for the position, and VCNL4010 distance sensor for the grasping action of the end effector.

- Allow the simulator to communicate accurately and effectively with the Intel Edison microcontroller. Once the data is formatted into the given format from the sponsor, the UDP packet is able to be send it to the simulator and the simulated Titan 4 arm must move according to the configuration variables being sent. There were some dropped packets during this phase, but a root cause was found to be the school's slow Wifi traffic.

- Complete a user setting button set-up to allow the user to increment how much control he or she wants of the arm. The settings include a lockout feature to lock the position and orientation of the arm, as well as the grasping of the end effector in case the user needs to take a break. In

Part Number	Quantity	Price/Unit	Total	Description
32150	2	$24.99	$49.98	propeller mini microcontroller
32201	1	$14.99	$14.99	prop plug to program microcontroller
90340	1	$6.27	$6.27	3/4" wide double sided velcro roll
160-1435-1-ND	10	$0.20	$1.96	Green LED 0603 2V 20mA
160-1436-1-ND	10	$0.20	$1.96	Red LED 0603 2V 20mA
160-1437-1-ND	10	$0.21	$2.11	Yellow LED 0603 2.1V 20mA
160-1646-1-ND	10	$0.39	$3.92	Blue LED 0603 3.3V 20mA
311-10.0KCRCT-ND	100	$0.01	$0.84	10k 1% 1/8W 0805
311-100KCRCT-ND	100	$0.01	$0.84	100k 1% 1/8W 0805
311-200CRCT-ND	100	$0.01	$0.84	200 ohm 1% 1/8W 0805
380-1006-ND	10	$0.43	$4.27	modular plug (RJ-45)
490-1536-1-ND	20	$0.03	$0.34	1uF 16V 0603 Ceramic Capacitor
609-4618-1-ND	5	$0.46	$2.30	surface mount microUSB connector
82-105-ND	1	18.23	18.23	60/40 rosin activated 26AWG solder (1/2 lb spool)
A122642-ND	10	$0.49	$4.91	modular plug boot
Arduino UNO	1	$9.99	$9.99	Arduino Uno Microcontroller
a15091800ux0023	1	$10.20	$10.20	double sided 100x70mm copper clad prototyping boards
B00HC98K2C	1	$13.50	$13.50	routing bits for making PC boards
B00NA3OMJO	1	$9.74	$9.74	2 inch wide elastic for arm bands
B00O2TWHWO	1	$9.99	$9.99	pack of 10 rectangular buckles for 2 inch straps
B00WEXKGH8	1	$7.99	$7.99	25 foot ethernet cable
BA033FP-E2CT-ND	3	$1.63	$4.89	3.3V Linear Voltage Regulator 1A
CN119B-500-ND	1	29.28	29.28	24AWG 500' silicone insulation 11/34 strand copper wire
CN119R-500-ND	1	29.28	29.28	24AWG 500' silicone insulation 11/34 strand copper wire
COM-12025	1	$19.95	$19.95	LED RGB strip (1 meter)
DEV-13097	4	$99.95	$399.80	Intel Edison Breakout Kit
ENC28J60	1	6.89	6.89	Sunkee Ethernet LAN Network Module
LSM303C	2	14.95	29.9	6 degrees of freedom Inertial Measurement Unit
LSM9DS1	2	24.95	49.9	9 degrees of freedom Inertial Measurement Unit
N/A	1	$3.32	$3.32	.3-1.2mm carbide bits for through-hole drilling
P12931SCT-ND	10	$0.37	$3.74	tactile switch surface mount 20mA
		Sub Total	**$752.12**	

TABLE I. TABLE 1 OF 2 SHOWING ALL PARTS PURCHASED DURING THE DESIGN AND TESTING OF THE PRECISION ROBOTIC MANIPULATOR CONTROLLER

this semester, the focus needed to be able to implement these features using basic push buttons with LEDs on the breadboard for the purposes of fulfilling a rapid prototype. These settings also allow the user to move the Titan 4 arm in quarter movements, half movements, and full movements.

Spring semester:

- Convert all Arduino and Propeller codes to the Intel Edison microcontroller. From the Fall semester, the Arduino, Propeller, and Intel Edison microcontrollers are used to process sensor data and communicate between them through serial. However, there needed to be code conversion from all of these to just the Edison microcontroller only as the sponsor mainly uses this microcontroller in their day-to-day operation. This way, all code is in one form and they can easily modify it to control other products after the completion of this project. The code is also separated into its own function calls within several files with detailed comments.

- Physical enclosure for the Intel Edison and breakout board needed to be designed and constructed. Several revisions were made to ensure great fit and attractive appearance. For the remote unit, a Wii nunchuck is as it contains the number of buttons and joysticks required by the secondary module. A breakout board was also designed and built as well to accommodate all of our features.

- The controller needed to be operable continuously for extended periods of time with no dropped packets. It also needed work with the simulator and the real Titan 4 arm. In order to accomplish this, safeguards in software were added to account for dropped packets or when

Part Number	Quantity	Price/Unit	Total	Description
PCE3868CT-ND	10	$0.73	$7.34	1000uF 10V Aluminum Capacitor
RJLSE4238101TCT-ND	5	$1.38	$6.92	modular connector (RJ-45) surface mount
S1012E-36-ND	10	1.403	14.03	0.1" pitch 36 position male header
S1112E-36-ND	10	1.964	19.64	0.1" pitch 36 position right angle male header
S7002-ND	10	0.463	4.63	0.1" pitch 4 position female receptacle
S7004-ND	10	0.528	5.28	0.1" pitch 6 position female receptacle
S7008-ND	10	0.675	6.75	0.1" pitch 10 position female receptacle
STY016011R	1	9.39	9.39	Stanley 16 inch tool box (7x16x7.5)
VCNL4010	2	$7.50	$15.00	Proximity / light sensor 1-10cm range
VL6180	2	24.95	$49.90	IR range finder with level converter (0-10 cm)
BNO055	5	$33.85	$169.25	Adafruit 9-DOF Absolute Orientation IMU
PI4ULS5V201TAEX	10	$0.57	$5.77	Voltage Level Translator Bidirectional
B0094X2066	1	5.69	$5.69	Wii nunchuck
TCA9548	1	6.95	$6.95	I2C multiplexer
COM-09032	2	5.95	$11.90	Sparkfun 2-axis joystick
36-941-ND	10	1.53	$15.30	micro USB connectors
LSM115JE3/TR13CT-ND	15	1.12	$16.80	0.22 V diode
MBRX120LF-TPMSCT-ND	5	0.49	$2.45	0.36 V diode
641-1697-1-ND	15	0.4	$6.00	0.40 V diode
MBR130T1GOSCT-ND	5	0.41	$2.05	0.45 V diode
S7012-ND	10	0.945	$9.45	14 pin female header 0.10"
507-1075-1-ND	5	0.318	$1.59	500 mA fuse 1206
311-0.0ARCT-ND	10	0.018	$0.18	0 ohm jumper 0805
399-11270-1-ND	5	2.37	$11.85	100uF 16v X5R 1210 capacitor
1276-1766-1-ND	10	0.022	$0.22	1pF 50 V capacitor 0805
399-5567-1-ND	10	0.264	$2.64	10pF 50V capacitor 0805
732-8046-1-ND	10	0.12	$1.20	100pF 25V capacitor 0805
AE10637-ND	5	0.56	$2.80	USB A connector - plug
B01M0W3WNF	1	6.88	$6.88	USB Charger / power supply
LYSB00VZXRRGG-CMPTRACCS	3	7.95	$23.85	Cat 6 Ethernet cable
		Sub Total	**$441.70**	
		Total	**$1,193.82**	

TABLE II. TABLE 2 OF 2 SHOWING ALL PARTS PURCHASED DURING THE DESIGN AND TESTING OF THE PRECISION ROBOTIC MANIPULATOR CONTROLLER

hydraulics are turned off and on. These safeguards are meant to prevent erratic behavior from the arm and crashing. This code was tested with the real Titan 4 arm and everything works as expected.

As needed to consider the design a success, all of the milestones needed to be, and were met.

VI. Project WBS and Task assignments

Table III represents the hours spent on the tasks set out in the Work Breakdown Structure (WBS). For a detailed description of the tasks themselves, see Section III.

Some of the tasks in Table III are also broken down into sub-tasks indicated with an arrow icon. Additionally, tasks such as 'Breakout Board' are not a total of its sub-tasks 'Circuit Design' and 'PC Board Design', hours logged in the parent task were still performed for that task, however the work did not break down specifically into the sub-tasks.

Hours logged into the 'Miscellaneous' task represent a combination of hours spent on conference calls, team meetings, research and writing of documents, project presentation and preparations, and other miscellaneous tasks. The 'Unused' row immediately below 'Miscellaneous' is an aggregation of work hours that went into parts and features that were ultimately either replaced with different parts and ideas or cut completely from the projects design.

The top two project tasks that required the most time to complete are the BNO055 Sensor and the Breakout Board at 244 and 130 hours respectively. Whereas the tasks that required the least time to complete are the creation of the Micro-Controller Enclosure and the assembly of the Main Controller at 13 and 11 hours respectively.

VII. Risk assessment and mitigation in WBS

At the beginning of the project, none of the members of the group had any experience working with IMUs, so they were a high-risk portion of the project. Additionally, they introduced other issues to the project that increased the level of risk of the project. After copious amounts of testing and a redesign of the project, we are now able to communicate to each of them using the single I2C bus available from the Intel Edison. A more detailed account of the troubleshooting efforts regarding the IMUs is given in Appendix B in the hardware testing section.

Initially, the Intel Edison itself was a high source of risk, considering that no members of the design group had used them before. Thankfully, programming on the Edison was not nearly as difficult as was anticipated. Instead of attempting to use Eclipse, the integrated development environment (IDE) that Intel suggests using for programming in C or C++, we use the GNU compiler GCC and program directly onto the Edison.

The distance sensor ended up not functioning within the specifications required by the project, or even within the specifications provided by the manufacturer, so a design change was necessary. Instead, control of the end effector was moved to the offhand module. After some discussion as to how to implement the module, a Nintendo Wii Nunchuk was used as the framework for our secondary module. Pressing up and down on the joystick controls opening and closing of the end effector, and left and right control its rotation.

VIII. Design overview

The design of the new controller is intuitive and easy to operate. To accomplish this, the design of the wearable device is light enough for the user to hold up their arm for extended periods of time without the weight hindering him or her. The wearable device is also made to be wearable by different sizes of people. The main controller weighs approximately 0.25 lbs, which is 0.25 lbs below the original goal. The controller is also designed to be easily adjustable as seen in the top of Figure 4.

Similar to the hardware and mechanical design of the wearable controller, the software is also clean and effective. Each feature has its own set of functions and files are separated properly. The code is commented in detail for readability purposes. This way, future programmers from TechnipFMC can easily implement this controller to other robotic arms besides the Titan 4. The software is also designed be easily modified to fit in new features and designs as well.

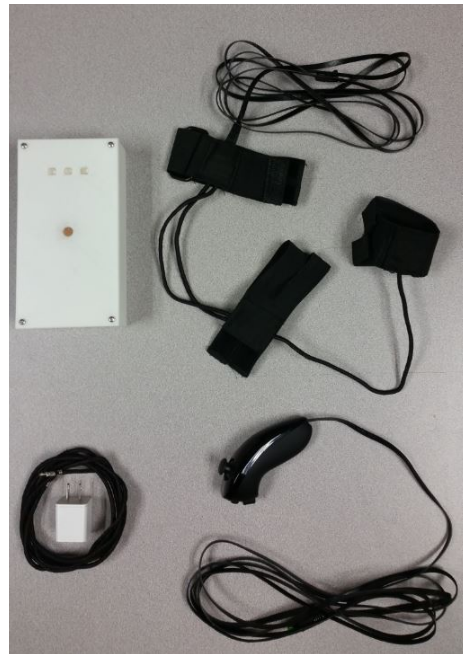

Fig. 4. Final controller, secondary controller, and enclosure designs.

In addition to a lightweight, adjustable main controller, a secondary module is also implemented and can be used with either hand as shown in bottom of Figure 4. It is a Wii nunchuck that is programmed to control the precise movements of the Titan 4 arm. It can control the wrist joint and the opening and closing of the end effector. Since these motions require precision, the joystick works very well since it is programmable to sense small and large stick movements. A lockout button and precision button is also incorporated into the nunchuck to add even more convenience for the user so they can take a break from operation without losing their progress, or reset their arm position to more easily manipulate the Titan 4 arm.

With the combination of these two main controllers under the processing power of the Intel Edison microcontroller, the result is a design that is practical and efficient and well within the goals of this venture.

IX. Deployable Prototype Status

A thorough test plan ensures the deployable prototype meets the design criteria and exceeds performance expectations by the delivery date. The full testing plan and timeline is shown in tables five and six of Appendix B - Hardware and Appendix C - Software. Based on this testing data, the deployable prototype meets all criteria specified in the design idea contract. The following results are a summary of the key components.

The device consists of three parts: the main controller, the remote unit, and the microprocessor enclosure. This is

Work Breakdown Structure tasks and hours					
Task	Amy	Charlie	Jamesson	Joseph	Total Task Hours:
Micro-Controller Enclosure	10	2	x	1	13
Breakout Board	x	88	12	1	101
-> Circuit Design	x	5	10	x	15
- >PC Board Design	x	4	10	x	14
Remote Unit	4	4	14	3.5	25.5
->Remote Unit Enclosure	x	2	9	x	11
->Remote Unit Conv. To Edison	x	x	12	2	14
Main Controller	x	8	x	3	11
Data Com. Coding Fram.	10	6	2	47	65
->Data Handling	x	2	x	19	21
->UDP Packet Structure	x	2	x	15	17
->Serial Data Input	1.5	1	x	7	9.5
BNO055	25.5	34	11	9	79.5
->User Hand Orientation	6	9	3	3	21
->Orientation Code Conv.	2.5	6	x	6	14.5
->Integration of all IMUs	3	39	4	3	49
->Position IMU Data	x	46	x	x	46
->Joint Variables from data	x	24	x	2	26
->Velocity Kinematics	2	2	2	2	8
Precision Control Feature	x	4	7	4	15
Lockout Feature	3	4	4	6	17
Status LED Programming	3	7	15.5	9	34.5
Miscellaneous	170	143	77	159	549
Unused	39.5	15	17	5	76.5
Total Team Member Hours:	280	457	209.5	306.5	1253
The **Unused** task relates to work with features and hardware that were replaced or cut from the final design					
The **Miscellaneous** task merges: calls, meetings, papers, presentations, preparations and misc tasks.					
**it should be noted that Jamesson tended to not include as many misc tasks as other teamates in Team activity reports and as a result has a lower total number of hours.*					
*** the '->' indicates a sub-task*					

TABLE III. HOURS SPENT ON WBS TASKS, INCLUDING RESPONSIBLE PARTY, TEAM MEMBER TOTALS AND TASK TOTALS.

verified visually using Figure 4 where all three parts plus the power connector are shown.

The data communication between the IMUs and the Intel Edison is now fully functioning. Problems arise when trying to connect multiple devices to the Intel Edison I2C bus because the Intel Edison does not have a strong low logic level. Two diodes are added to increase the ground potential of the IMUs to the low logic level of the Edison and 6 additional diodes are added reversed biased between the data lines and ground, one per each data line and one per clock line to each IMU. The addition of these diodes allows for a reliable connection between the BNO055 IMU and the Intel Edison Microprocessor. The test of this feature was completed on 3/11/2017.

The standard movement of the robotic manipulator based on user arm movement is implemented and final testing for this general feature completed on 4/19/2017 when the physical Titan 4 manipulator was successfully controlled using the manipulator.

The lockout function is vital before any testing can start otherwise any user arm movement will move the robotic manipulator. Testing for this was officially completed on 3/11/2017 even though it was functioning before that.

Position Reset is fully functioning and when a user uses the lockout, the controller comes out of it with the user's current arm position corresponding to the current manipulator position. This is included in the testing for lockout resume and was completed on 4/17/2017.

The UDP data transfer, specifically to a physical Titan 4 arm, is one of the last features tested due to access to the Titan 4 arm. This test was completed successfully on 4/19/2017 after some initial minor problems delayed the success.

The ease of use testing is not specifically listed in the testing plan and timeline table but it is important to note that the control of the physical Titan 4 arm performed on 4/19/2017 was done by multiple users including a TechnipFMC employee that did not use the controller prior to that day. All users were able to stack pieces of metal using the controller, demonstrating the ease of use for first time users.

More specific details on the smaller features are located in the Hardware and Software sections of Appendix B and Appendix C. All tests pass, the deployable prototype performs all required tasks, and it includes all required features as specified in the design idea contract.

X. Deployable Prototype Marketability Forecast

The end product produced from both the Fall and Spring semesters is a controller set that is reliable and easy to use. In terms of having a deployable prototype, this product meets all of the project requirements set in the beginning of the semester.

The software is modularized and commented so FMC can easily understand all of the code and adjust it to work with multiple robotic arms. As it stands, the software side of the project is very reliable and is able to control the Titan 4 arm successfully. The one improvement would be to add in more safe guards for any unpredicted failures or if the sponsor would like to use this controller on a moving ship, the IMUs will need some software tweaks to work with moving water surface. Besides these improvements, this software is ready to be deployed.

In terms of physical appearance and hardware of the controller set, a professional enclosure should be made and a professional PCB breakout board should be created to replace the current hardware. This will increase the product's market value and durability. This project's hardware functionality is all working properly and is ready to be deployed to FMC's customers otherwise.

All in all, this product is very marketable with very little changes needed to be fully deployable.

XI. Conclusion

These past two semesters of senior design have been a stress-filled learning experience. A difficult project concept was chosen to challenge the abilities of the groups engineering skills. Throughout the first semester, a rapid prototype was produced with little difficulties due to the utilization of microcontrollers that were familiar and simple to implement. However, second semester proved to be three times as challenging because of a new microcontroller implementation and integrating of all the working parts into one. With great effort and time spent on this project, integration of all components into one working piece was accomplished and a controller was produced that exceeds all engineering objectives set by the design idea contract and sponsors. This project was appropriately labeled a success and moving forward the group plans to continue to develop valuable skills learned here like teamwork, engineering thinking, and time management.

XII. References

Bosch, "Features," BNO055 datasheet, Jun. 2016 [Version 1.6].

Intel, "Introduction", Intel Edison Board Support Package, Feb. 2015, [Revision 005].

Vishay, "Features," VCNL4010 datasheet, Aug. 2014 [Revision 1.6].

Intel, "High Level Functional Description," Edison Compute Module datasheet, Jan. 2015 [Revision 004].

XIII. Glossary

Accelerometer: an instrument for measuring acceleration of objects.
Arduino: a brand of electronics manufacturer, known for, among other things, making multiple models of microcontrollers.
Breakout Board: a surface-mount chip soldered onto a PCB to make prototyping easier.
Cartesian Coordinate System: a coordinate system that specifies each point uniquely in a plane by two or three numerical coordinates, which are the signed distances to the point from two or three fixed perpendicular directed lines, measured in the same unit of length.
Degree of Freedom (DOF): a direction in which independent motion can occur.
End Effector: a device at the end of a robotic arm, designed to interact with the environment. The exact nature of this device depends on the application of the robot and the task to be completed.

Gyroscope: a device consisting of a wheel or disk mounted so that it can spin rapidly about an axis that is itself free to alter in any direction. In a sensor, it detects orientation of the object.
Inter Integrated Circuit (I2C): a multi-master, multi-slave, single-ended, serial computer bus that allows multiple slave chips to communicate with one or more master chips.
Integrated Developement Environment (IDE): software application that provides comprehensive facilities to computer programmers for software development.
Inertial Measurement Unit (IMU): a self-contained system that measures linear and angular motion usually with a triad of gyroscopes and/or triad of accelerometers.
User Datagram Protocol (UDP): an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss tolerating connections between applications on the Internet.
Intel Edison: a tiny computer-on-module offered by Intel as a development system for wearable devices and Internet of Things devices.
Magnetometer: an instrument used for measuring magnetic forces, especially the earth's magnetism.
Parallax Propeller: microcontroller designed to perform multiple tasks simultaneously, and without the need for interrupts or the dictates of an onboard operating system. Each of the Propeller P8X32A's eight symmetrical cores can access all 32 I/O pins and other shared system resources. can be programmed in Spin or C.
Potentiometer: an instrument for measuring an electromotive force by balancing it against the potential difference produced by passing a known current through a known variable resistance.
Printed Circuit Board (PCB): an electronic circuit consisting of thin strips of copper that have been etched from a thin sheet of copper on a laminate in a specific configuration so integrated circuit components can be connected in a useful manner.
Red Green Blue Light Emitting Diode (RGB LED): combine these three colors of Red, Green, and Blue to produce over 16 million hues of light.

XIV. Appendix A. User Manual

The following documentation shows the user manual created for this project.

Precision Robotic Arm Controller

User Manual

This manual serves to inform the user how to operate the precision robotic arm controller, designed and constructed by Team 2 of the joint Electrical and Electronic Engineering and Computer Engineering Senior Design class of California State University, Sacramento, College of Engineering and Computer Science from Fall 2016 to Spring 2017. This project was sponsored and funded by TechnipFMC in Davis, CA.

Contents

Glossary of Terms

- End effector – end of the robotic arm, for performing specific tasks.
- Inertial Measurement Unit (IMU) – electronic sensor, capable of determining force, angular rate of change, and strength of magnetic field. The basis for this design.
- Microcontroller – a small computer, usually for performing a single task or group of small tasks.
- RJ-45 – a type of electrical connector, typically associated with Ethernet cables.

Components

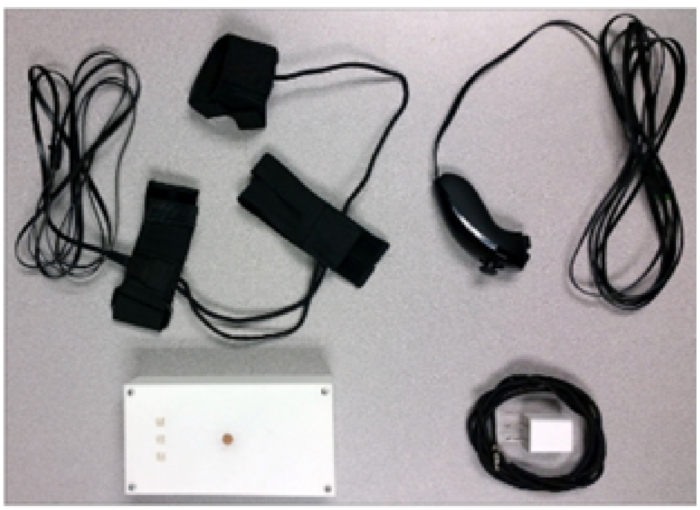

Fig. A-1: Different components of the final design.

The different main components of this project, as depicted in Figure A-1, are as follows, starting in the top-left and moving clockwise:

1. Main controller – consists of three Velcro bands into which have been inserted the three IMUs that the controller uses. These straps are in turn placed on the bicep, forearm, and the back of the hand. The one closest to the shoulder includes an RJ-45 port; Ethernet cable is provided to connect the main controller to the enclosure.
2. Offhand module – Controller for the end effector. Also allows the user to make changes to the various settings of the controller.

3. Power – The controller requires up to 2 amps at 5 volts. Power is supplied via a type A to micro-USB cable.
4. Enclosure – protective box that houses the microcontroller that handles the controller's actual program.

Theory of Operation

Note: this manual assumes that the arm has been properly set up and is ready to use.

1. Strap the main controller onto the arm: one section straps to the back of the hand, one in between the wrist and the elbow, and one in between the elbow and the shoulder. Note: the section on the shoulder needs to be on top of the shoulder, or the controller will not function as desired. Additionally, if the user's arm is extended fully, the IMU sensors should all face directly upward.
2. Plug the power adapter into the wall.
3. Plug micro-USB cord into the power adapter.
4. Plug the micro-USB cord into the micro-USB port of the enclosure. Note: the microcontroller is a small computer: it will need a few moments to power on and start up.
5. Wait until the three LEDs on the enclosure turn on; they will be pink at this point.
6. Connect one end of an Ethernet cable to the shoulder section of the main controller.
7. Connect the other end to the port on the enclosure labeled "Controller".
8. At this point, the LEDs will turn yellow.
9. Wave the arm with the controller on around until all three LEDs turn back to pink, at which point the LEDs will begin normal operation according to Table A-1:
10. Table A-1 below depicts the error information for each LED, starting from left to right.

Error LED	Precision LED	Lockout LED
IMU not Initialized	1:1 Movement	Fully Locked
IMU not Calibrated	2:1 Movement	Fully Unlocked
Beyond Reach	4:1 Movement	No Jaw Movement
Invalid Hand Config		
Found by Iteration		
No Errors		

Table A-1: Color chart for each of the LEDs on the enclosure. The error messages listed are defined in Troubleshooting.

11. Starting LED configuration should be Teal, Sky Blue, Red.
12. Press both buttons on the offhand module simultaneously to toggle the hydraulics of the arm. The controller turns the hydraulics on at startup.

The arm is now ready to use. When the task is complete and the arm needs to be stowed, press the circular button located on the center of the enclosure's lid. This will lock out the arm against user movement and reposition the arm to a pre-determined location.

Offhand Module

Figure A-2: The offhand module for the precision controller.

End Effector Controls

The joystick of the offhand module controls the end effector. Pressing up on the joystick will open the jaws. Pressing down will close them. Similarly, pressing right will rotate the end effector clockwise relative to the user, and pressing left will rotate it counter-clockwise relative to the user. Note: there are two movement modes no matter in which direction the joystick is pressed. In other words, pressing the joystick a little bit in any direction will cause less movement in that direction than pressing the joystick fully in the same direction.

Settings

There are two buttons on the offhand module. Pressing them simultaneously turns on/turns off the hydraulics to the arm. Pressing the top button alone cycles through the different lockout settings, as depicted in the right column of Table A-1:

- Red – the program defaults to Fully Locked when the controller is initialized, signified by a red light. This means that, no matter what the user does with their arms, the robot arm will not move.
- Green – Fully Unlocked. This means that the arm is fully capable of moving, including the end effector. Additionally, whatever position and orientation of the hand and arm of the user when the system transitions from Fully Locked to Fully Unlocked will be the starting position of the robot arm. In other words, movement of the robot arm is relative to the starting position of the arm when the

transition is made.

- Yellow – No Jaw Movement. This means that the arm is capable of full movement, and the end effector is capable of rotating, but the jaws will not open or close.

Pressing the lower button cycles through the sensitivity settings, shown in the middle column of Table A-1:

- Sky Blue – This light corresponds to 1:1 Movement. This means that any angular movement of the user's arm will correspond to the same amount of angular movement of the robot arm.
- Dark Blue – 1:2 Movement. Any angular movement of the user's arm corresponds to half of that angular movement of the robot arm.
- Violet – 1:4 movement. Any angular movement of the user's arm corresponds to a quarter of that angular movement of the robot arm.

Troubleshooting

If the arm is not moving, use the following table to determine the cause of the issue.

LED Color	Potential Problem	Solution
No light/all are black	Controller not plugged in	Plug in appropriate cable (power, Ethernet, etc.)
All are pink	IMUs not initialized	Unplug and plug in main controller Ethernet cable
All are orange	IMUs not calibrated	Wave arm around until no LED is orange
All are different colors, lockout LED stuck on red	Hydraulics are not on	Press both buttons simultaneously
Lockout LED is red	Arm is fully locked	Press top button once on offhand module
Error LED is lavender	Trying to reach farther than arm is capable	Lock arm, then extend main controller arm, then unlock arm
Error LED is red	Arm is in an invalid position	Slowly move individual joints of the arm until robot arm moves; bend hand at the wrist back and forth; lock arm, then unlock arm.

XV. APPENDIX B. HARDWARE

The following documentation shows the hardware block diagram, schematics for the three circuit boards created for this project, hardware used in the final prototype and the specific tests relating to the hardware for the device.

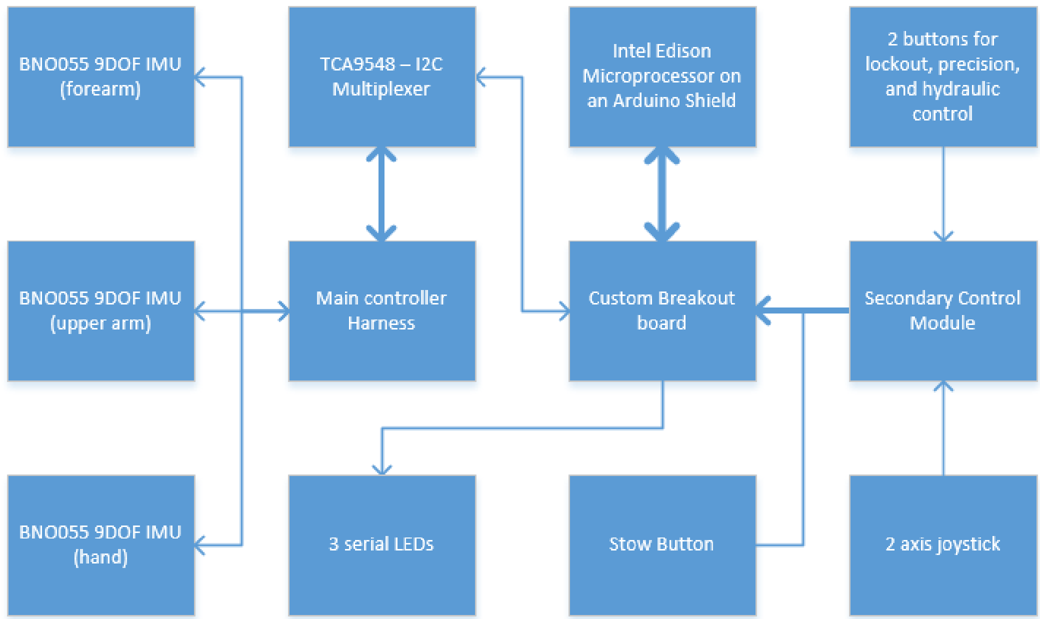

Fig. 5. flowchart showing all of the electronic hardware components and interconnections

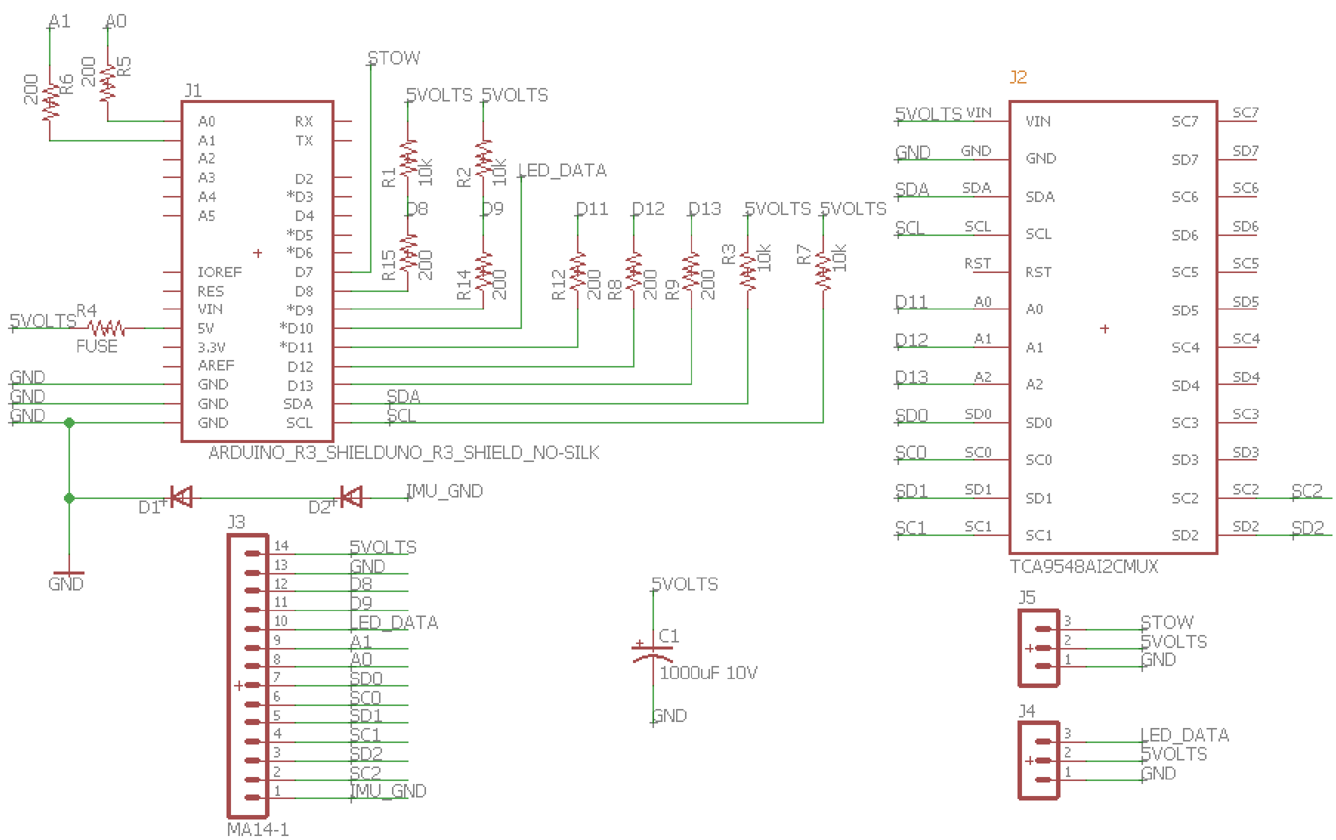

Fig. 6. Schematic of the main breakout board, connected directly to the arduino shield

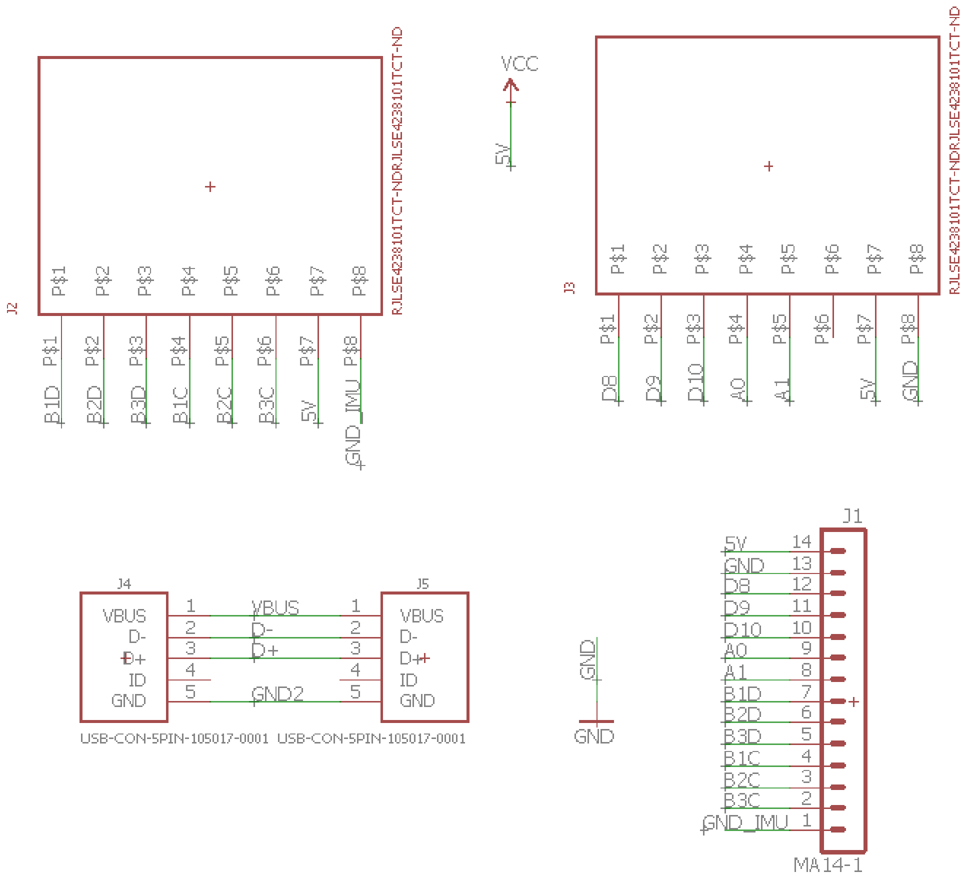

Fig. 7. Schematic of the connector board inside the enclosure, provides connection to the main controller and remote unit from the Edison through the 14 pin header

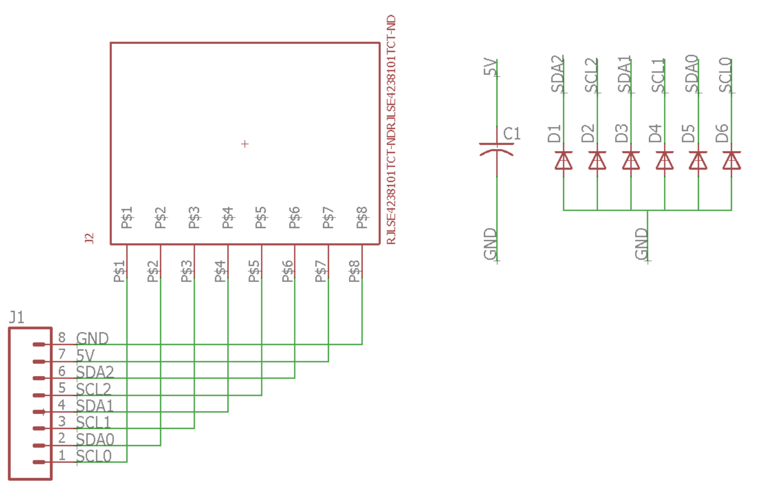

Fig. 8. Schematic of the small circuit board in the main controller, containing connections for the IMUs and also the clipping diodes for the data lines

Part Number	Quantity	Price/Unit	Total	Description
90340	0.5	$6.27	$3.14	3/4" wide double sided velcro roll
311-10.0KCRCT-ND	20	$0.01	$0.20	10k 1% 1/8W 0805
311-100KCRCT-ND	5	$0.01	$0.05	100k 1% 1/8W 0805
311-200CRCT-ND	5	$0.01	$0.05	200 ohm 1% 1/8W 0805
609-4618-1-ND	2	$0.46	$0.92	surface mount microUSB connector
82-105-ND	0.0625	18.23	$1.14	60/40 rosin activated 26AWG solder (1/2 lb spool)
a15091800ux0023	0.2	$10.20	$2.04	double sided 100x70mm copper clad prototyping boards
B00HC98K2C	0.25	$13.50	$3.38	routing bits for making PC boards
B00NA3OMJO	0.3	$9.74	$2.92	2 inch wide elastic for arm bands
B00O2TWHWO	0.1	$9.99	$1.00	pack of 10 rectangular buckles for 2 inch straps
COM-12025	0.05	$19.95	$1.00	LED RGB strip (1 meter)
DEV-13097	1	$99.95	$99.95	Intel Edison Breakout Kit
N/A	1	$3.32	$3.32	.3-1.2mm carbide bits for through-hole drilling
P12931SCT-ND	1	$0.37	$0.37	tactile switch surface mount 20mA
PCE3868CT-ND	1	$0.73	$0.73	1000uF 10V Aluminum Capacitor
RJLSE4238101TCT-ND	3	$1.38	$4.14	modular connector (RJ-45) surface mount
S1012E-36-ND	2	1.403	$2.81	0.1" pitch 36 position male header
S1112E-36-ND	2	1.964	$3.93	0.1" pitch 36 position right angle male header
S7002-ND	2	0.463	$0.93	0.1" pitch 4 position female receptacle
BNO055	3	$33.85	$101.55	Adafruit 9-DOF Absolute Orientation IMU
B0094X2066	1	5.69	$5.69	Wii nunchuck
TCA9548	1	6.95	$6.95	I2C multiplexer
36-941-ND	3	1.53	$4.59	micro USB connectors
LSM115JE3/TR13CT-ND	7	1.12	$7.84	0.22 V diode
641-1697-1-ND	1	0.4	$0.40	0.40 V diode
S7012-ND	2	0.945	$1.89	14 pin female header 0.10"
311-0.0ARCT-ND	2	0.018	$0.04	0 ohm jumper 0805
399-11270-1-ND	2	2.37	$4.74	100uF 16v X5R 1210 capacitor
AE10637-ND	1	0.56	$0.56	USB A connector - plug
B01M0W3WNF	1	6.88	$6.88	USB Charger / power supply
LYSB00VZXRRGG-CMPTRACCS	2	7.95	$15.90	Cat 6 Ethernet cable
		Total	**$289.02**	

TABLE IV. LIST OF THE PARTS AND QUANTITIES USED FOR IN THE FINAL DESIGN OF THE DEPLOYABLE PROTOTYPE

HARDWARE TESTING PLAN							
Test ID	Description	Expected Results	Actual Results	Pass/ Fail	Additional Comments	Completion date	Team Member assigned
007	Reasonable setup/removal time	User can start using robotic controller in < 3 minutes	Easily able to setup and begin device usage within a three minute time period	PASS	User is able to fully setup both the Enclosure and the wearable portion withing the alloted time.	3/11/2017	Charlie
012	BNO055 Verification	Accurate and stable orientation values from the sensors	The BNO055 sensors are able to accurately read calculate and send stable data.	PASS	The multiplexer circuitry is all functioning for this data to arrive successfully	3/11/2017	Amy/ Charlie
015	Breakout Board testing	Multimeter tests; make sure all wires and traces are correct	All traces are independent of other traces when tested with a multimeter	PASS	All three custom circuit boards are tested and all problems fixed for proper signal transmission	3/1/17-3/11/17	Jamesson/ Charlie
016	Remote Unit Testing	Verify all buttons work as intended; LED's display as intended	Able to detect when each button is being pressed, independently and together	PASS	LEDs are no longer part of the remote unit but they also work	3/11/2017	Jamesson

TABLE V. TESTS PERFORMED FOR BOTH HARDWARE AND SOFTWARE TO ENSURE PROPER FUNCTIONALITY.

XVI. Appendix C. Software

The following appendix documentation shows the software block diagram, flowchart, pseudo code and device testing with the results related to the software portions of this project.

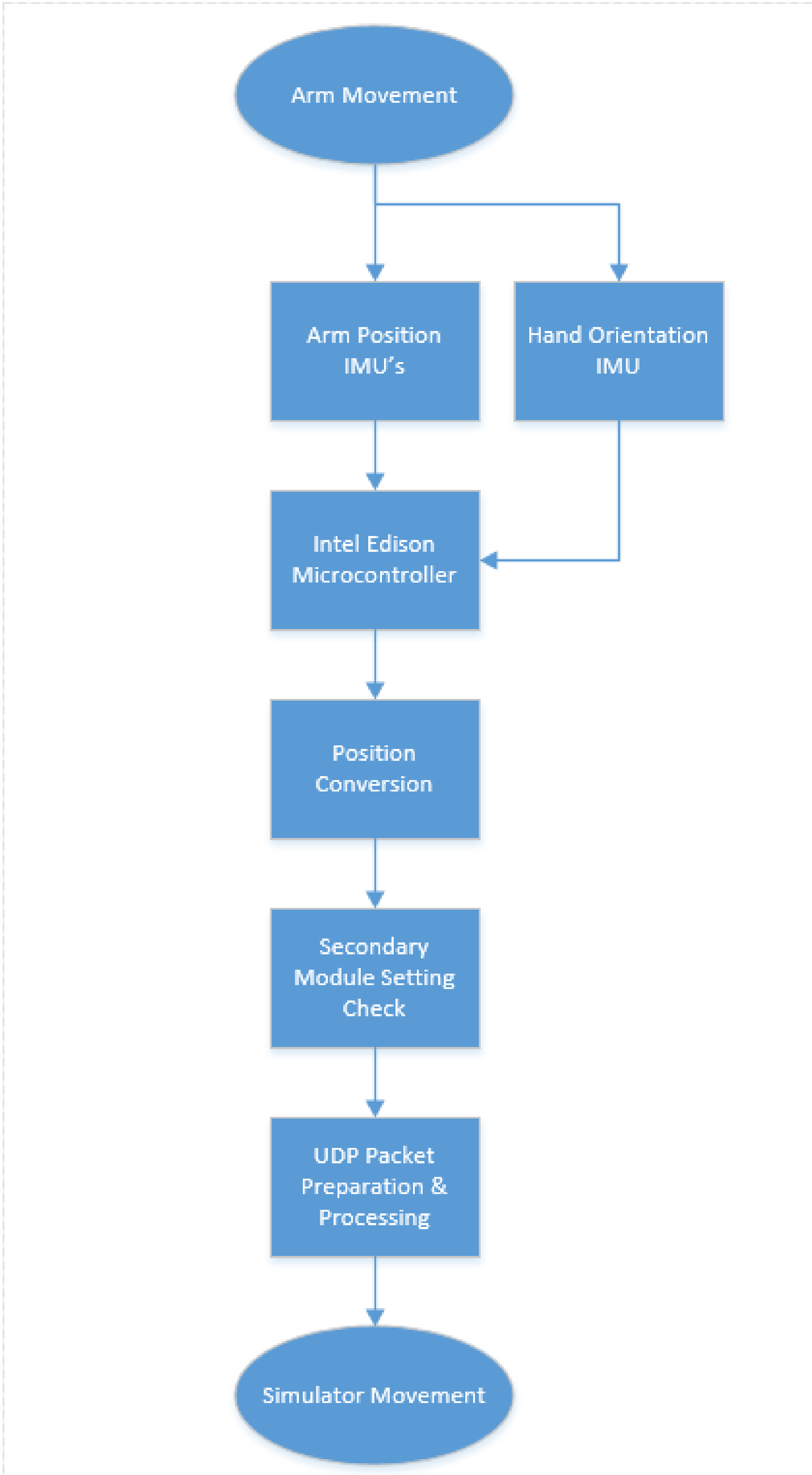

Fig. 9. Software block diagram for Precision Controller.

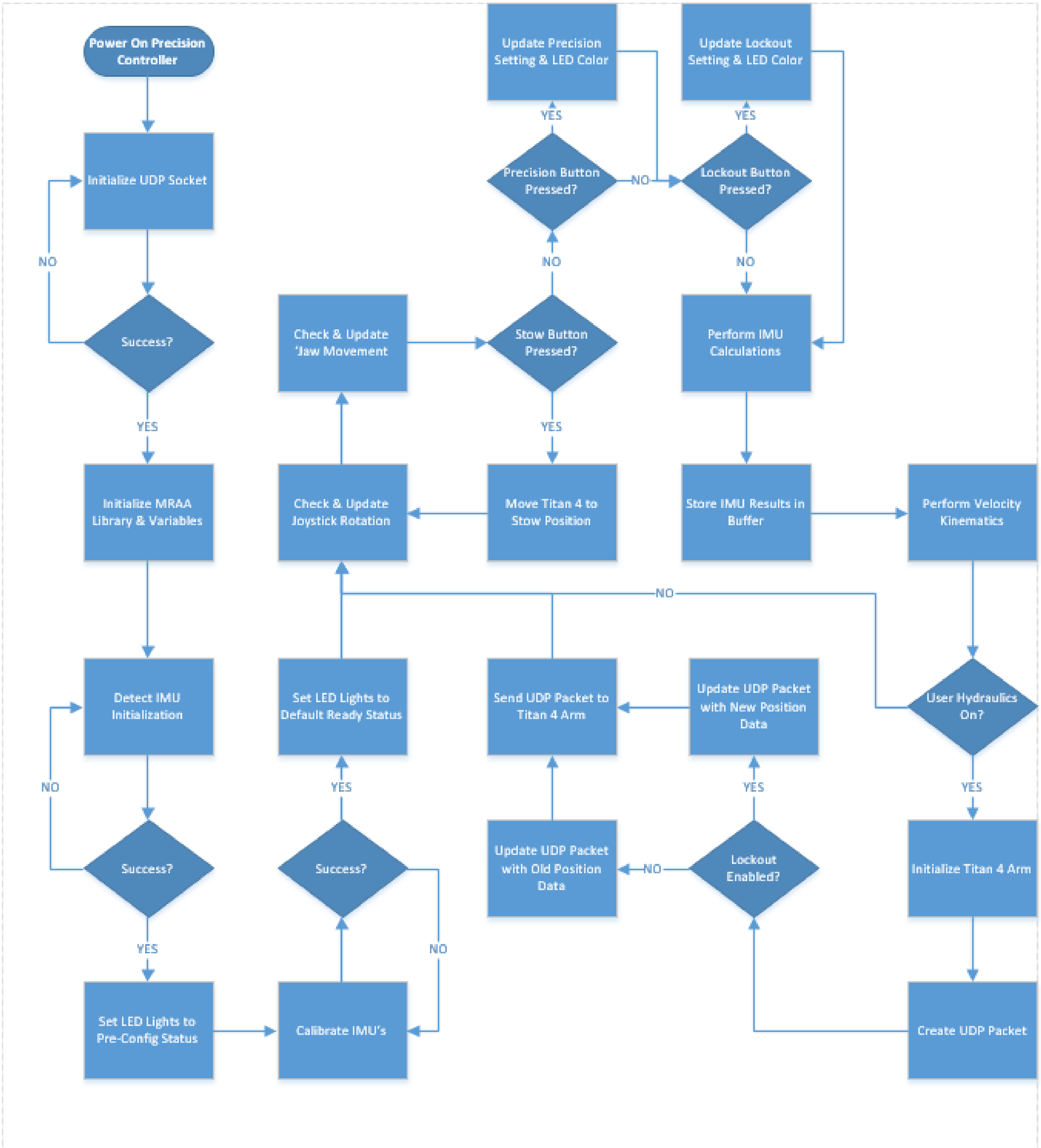

Fig. 10. Software flowchart for the Precision Controller.

```
/* Psudocode for Precision Robotic Manipulator
Controller */
-Create a UDP socket to communicate with the Titan 4
Hydraulic arm

-Initialize necessary pins and buttons for user input

-While the IMU's are not initialized loop through and
initialize all three IMU sensors
        -Then Update LED colors

-While the sensors are not calibrated, run calibration
functions for the IMU sensors
        -Then Update LED colors

-Forever loop
        -Check for User input
        -If a input is detected on the joystick
                -Update rotation/Jaw value

        -If one of the buttons were pressed
                -Wait for a time
                -Recheck the buttons again to determine
                if it was a 2 button press

        -If the precision and lockout button is pressed
                -Toggle the hydraulics variable
                -Update LED colors

        -If only the precision button was pressed
                -Change precision state
                -Update LED color based on state

        -If the lockout button was pressed
                -If Hydraulics are on
                        -Change lockout state

                -If state 0
                        -Update color
                -Else If state 1
                                -Update LED color
                                -Get the most recent
                                position data
                                -Update the buffer
                                -Set the current and
                                recent buffers
                -Else If state 2
                                -Update LED colors

        -Else If the stow button is pressed
                        -Change to lockout
                        -Update the buffer
                        -Update the stow buffer
                        -Update LED colors

        -If not locked out and Hydraulics are on
                -Update last position buffer
                -Pull new positions from IMU's
                -Calculate new variable positions

                -Check to see if they are out of range
                        -If so, set the current position to
                        the last position
                        -Update the LED colors

                -Else if it is in range
                        -Update the LED colors
                        -Check to see if it's a
                        valid configuration
                        -If not, set current position
                        to last position
                        -If it is valid
                          -Change LED colors
                          -Change new position to
                          the calculated variables

        -Move the valid position information into the
        UDP buffer

        -If Hydraulics are on and it's not locked out
                -Perform the movement using velocity
                kinematics

        -If hydraulics is on and the arm is not initialized
                 -Run the initialization sequence
        -Else if hydraulics are on and the arm is
        initialized
                -Validate UDP packet data
                -Build UDP packet with checksum etc.
                then append position information

                -If the UDP communications are valid
                                -Send the UDP packet

                -If the UDP communications are valid
                                -Receive the reply

                -Continue the forever loop based on
                reply information

-End forever loop
```

Fig. 11. Software pseudo code for the Precision Controller.

SOFTWARE TESTING PLAN

Test ID	Description	Expected Results	Actual Results	Pass/ Fail	Additional Comments	Completion date	Team Member assigned
001	Max/Min angles within range	Possible joint values not to exceed Max/Min valid range	All joints tested independently and limits before max angles exist	PASS	All jumping has been fixed and no invalid movement is possible	4/15/2017	Charlie
002	Continuous rotation of wrist joint	Wrist joint rotates indefinitely, left or right, based on input	Joystick accurately controls continuous wrist rotation based on two distinct speeds	PASS	Rotation works correctly for both left and right rotations.	3/11/2017	Amy/ Jamesson
003	No runtime errors/freezing	Simulator runs for > 12 hours with no errors	The Simulator runs for over 12 straight hours, occasionally moving the controller to ensure functionality	PASS	initial freezing issues are resolved with dropped packet handling code	3/18/2017	Joseph/ Charlie
004	Lockout Enable working	Activation of lockout feature stops position transfer	Lockout button correctly cycles through lockout sequence and correctly disables arm segments	PASS	3 lockout steps work correctly, button sequence is correct and works continuously without error	3/11/2017	Jamesson
005	Lockout Resume working	When lockout is disabled, arm moves proportional to user	Lockout resume button correctly re-engages robotic manipulator in the simulation	PASS	The simulator is able to resume on command with no apparent issues.	4/17/2017	Jamesson
006	Precision Control functionality	When enabled, the robot arm moves about half as far as the user's hand	Double and 4 times precision modes both work as anticipated	PASS	Checked using camera on z-axis of simulator and rotating 90 degrees	4/10/2017	Jamesson
008	Prototype resumes after power loss	After power loss, Prototype resumes functionality automatically	Loss of power resulted in a reboot and the Micro-controller properly started after boot up	PASS	Re-establishing power causes the device to power up and automatically initialize the program.	2/24/2017	Joseph/ Amy
009	Jaw opening/closing is accurate	The jaws open when the joystick is moved up and close when the joystick is moved down	Joystick accurately controls jaw direction on the physical Titan 4 arm	PASS	The joystick accurately controls both opening and closing of the jaws based on the two ranges of opening and closing speed.	4/19/2017	Jamesson/ Charlie
010	Data handling reliability	Ensure that data is received and stored correctly	Position data is received, stored, and sent back out to the arm without jumping	PASS	The data is received reliably otherwise the arm would jump when sending the received position back to it	4/10/2017	Joseph
011	Data communication coding framework	Ensure data sent to arm is correct, and errors are handled as needed	Data is able to be sent to and received from a physical Titan 4 manipulator	PASS	The testing with the real arm verified communication works between the two	4/19/2017	Joseph
013	Velocity Kinematics code testing	Output values are calculated to move at a constant velocity over time	The values of the joints step to the desired value, viewed with the console	PASS	the velocity kinematics have eliminated any instantaneous jumps	4/10/2017	Amy
014	Velocity Kinematics arm test	Verify user to robot arm movement ratio is within valid range	The physical arm does not move faster than the limit with quick movements	PASS	Test performed on the simulator and then verified on the physical manipulator	4/19/2017	Amy

TABLE VI. TESTS PERFORMED FOR THE SOFTWARE TO ENSURE PROPER FUNCTIONALITY.

XVII. Appendix D. Mechanical

The following documentation shows the mechanical designs for the 3D printed enclosure. The enclosure is made up of a rectangular base with mounting poles for the Intel Edison and the printed circuit boards as well as cutouts to connect to the Main Controller and Secondary Module.

The lid of the enclosure allows for four screws at the corners to fasten the box closed. Additionally, there is a cutout for the center stow button and three small squares used for LED placement.

Box exterior dimensions: 180mm by 100 mm 50 mm.

Fig. 12. Lid design for the Precision Controller enclosure.

Fig. 13. Base design for the Precision Controller enclosure.

XVIII. Appendix E. Vendor Contacts

Kevin Hjelden from TechnipFMC is an excellent resource for suggestions, Titan 4 interface protocols and parts acquisition from distributors. There is no direct communication between any team member and any other vendors due to the nature of the project.

XIX. Appendix F. Resumes

This section contains the resumes of all group members as of the end of the project.

Amy M. Giovinazzo

ELECTRICAL ENGINEERING · DESIGN

Education

Bachelor of Science in Electrical Engineering — ***CSUS GPA: 3.9***

Sacramento State University — ***Graduation: May 2017***

- Objective: to gain a full-time position with an engineering firm that will offer competitive pay, growth and work-life balance.

Skills

SUMMARY

- Languages: fluent in Thai and English.
- Technical: Exposure to Python script, Bash script, C, Java, Linux, x86 Assembly, Verilog, MATLAB, and PSPICE.
- Personal: Excellent communication skills, a fast-learner, work well within a team and individually, great attention to detail, highly organized, and a multi-tasker.

Experience

Senior Design — *CSUS*

FMC Schilling Robotics sponsor — *Fall 2016-Spring 2017*

- Design and implement a controller, based on human arm movements that will allow for a more intuitive and precise control of the Titan 4 robotic arm. Project sponsored by FMC Schilling Robotics.

DCPAE - Platform Application Engineering Intern — *Folsom, CA*

Intel Coporation — *May 2016 - present*

- In charge of updating and testing for firmware BKCs for internal and external Grantley and Purley systems.
- Created Python script to parse current generation of error codes plus its subcategories, and automated csv output that can be given to customers. Changed a manual process into an automated one. Process was integrated into future ACM releases.
- Cut down wait time and increased efficiency of image generation by creating a remote imaging server with Clonezilla that enabled remote clone, upload and download of different OS images.
- Enabled engineers to internally debug customer issue with SVOS by figuring out how SVOS works and configuring the server for step testing. Seeked out resources to complete the task.

Software Engineering Intern - Networking — *Roseville, CA*

Hewlett-Packard Enterprise — *Sep. 2015 - Jan. 2016*

- Implement various debugging techniques, including printf debugging, commenting/uncommenting code debugging, and A-B-A comparisons.
- Worked in Google test environment to find bugs in switch firmware/software using C and C++.
- Created code reviews in Collaborator and incorporated code feedback to improve submitted code.
- Worked on Customer Requests (CRs) to fix bugs for release cycle for HP 3810 Fixed Port L3 Managed Ethernet Switches in C. Used Multi debugger to step through code. Increased Google test to 89% lines and 98.9% functions coverage tested.

Hardware Engineering Intern - Router — *Grass Valley, CA*

Grass Valley Group, A Belden Brand — *May 2015 - Aug. 2015*

- Tested baseline Alignment Jitter and Timing Jitter by isolating the FPGA system and bypassing the FPGA (equalizer board with jitter cleaner with register microcontroller, 2 different types of reclocker boards, and cable driver) to see how the video signal behaves. Successful testing will win Sky Brazil's possible $10 million contract.
- Documented all data and proposed the best solution to the sales team so they can effectively communicate with Sky Brazil and overcome objections.
- Got familiar with Grass Valley's equipment like switchers, routers in hardware, software, and IP.

Cars Internet/Sales Consultant — *Tracy, CA*

American Chevrolet, Tracy Mazda, Porter Hyundai — *Aug. 2011 - May 2015*

- Consistently obtained average of 14 cars/month and top salesperson in most months.
- Managed existing customers who have not purchased, and handle all incoming recall appointments as well as ordering of parts needed for the recalls.

R & D Intern — *Livermore, CA*

Lawrence Livermore National Laboratory — *Summer 2010 and Summer 2011*

- Interned with Dr. Larry Fried on the Cheetah program. Researched chemical elements and its properties and how they react to one another in the Cheetah engine simulations.
- Shadowed Dr. Sat Pannu on the Vision project. Obtained a Secret Clearance.

Extracurricular & Honors

CLUBS & ORGANIZATIONS

- **Society of Women Engineers 2016-2017:** Rewarded the Sacramento County scholarship.
- **Competitive Robotics (Event Coordinator, Industry Liason 2015-present)**
- **Tau Beta Pi (present):** Sponsor Chair elect.
- **American Indian Science and Engineering Society (AISES) 2015-2016:** member
- **High Power Rifle Team (Secretary 2009-2011):** Organized each practice and made sure that everyone practiced safe shooting/target pulling techniques, used M-16s to practice at 500 yards with iron sights. Expert in rifle and pistol Navy.

Joseph D. Smith

Summary

Experienced and reliable with a strong technical background and excellent client service communications. Versatile and able to quickly adapt to fluid project goals. Experienced with various programming platforms.

Highlights

- ❖ Experience with C/C++, C#, Verilog, VBA, X86 assembly, MATLAB, Cadence software & GIT
- ❖ 1 year stimulating, simulating and emulating test models in a Windows environment
- ❖ Strong evaluation skills; including measurements and analysis
- ❖ 4 years of computer diagnostics and repairs in work settings
- ❖ 5+ years reviewing, analyzing and documenting operational procedures of varying nature
- ❖ Multi-tasking and self-starter skills with ability to understand difficult concepts and solve problems
- ❖ Possess SECRET security clearance

Education

Bachelor of Science, **Computer Engineering**
California State University, Sacramento
Expected Date of Graduation: May 20th, 2017
CSUS GPA: 3.3

Professional Work Experience

TechnipFMC (Senior Design) Aug 2016 – May 2017
Student Contractor

- Sponsored by TechnipFMC to design and implement a controller, based on human arm movements that will allow for a more intuitive and precise control of the Titan 4 robotic arm.

Space and Naval Warfare Systems Center (SPAWAR) May 2013 – Sept 2014
Security System Support Specialist

- Analyzed and documented operational procedures related to electronics configuration and implementation.
- Built, assembled, and tested electronic computer systems including server rack systems.
- Conducted equipment studies and gathered market research data regarding large scale upgrade options.
- Measured, analyzed and tested virtual machine models using simulation, stimulation and/or emulation.
- Configured, updated and installed Cisco Network switches at SPAWAR sites

Projects:

- Installed and maintained Windows Server in a test environment, including the management of client connections and privileges. Hardened windows server and clients against security vulnerabilities.
- Created an automated document using Microsoft VBA. The document searched through hundreds of rows of data across multiple sheets to indicate the status of the necessary checks on equipment at SPAWAR sites. Additional features were added to the program to auto save the results, as well as restricting fields and options from non-administrative users.

Staples Nov 2009 – May 2013
Resident Computer Technician

- Intake and assessment of customers' computers and their associated problems.
- Well versed in computer hardware and software diagnostics.
- Replacement of computer parts to solve hardware malfunctions.
- Pride in my ability to surpass my customers' expectations concerning repairs and timelines.

CHARLES RELAT

Electrical and Electronic Engineering student graduating in May, 2017 with a primary focus on control systems and secondary focus on power systems with more than 5 years of professional experience troubleshooting/testing electronic control and communications products and systems.

EDUCATION

California State University, Sacramento — exp. 2017

Bachelor of Science Electrical and Electronic Engineering — GPA: 3.9

- Active member of Tau Beta Pi
- Member of IEEE – Sacramento Valley Section

American River College — 2013

Associate of Science Electronic Systems Technology — GPA: 3.4

- I.S.C.E.T. certification
- F.C.C. General Radio Operators License
- First place nationally in the Skills USA Electronics Technology Competition

PROFESSIONAL EXPERIENCE

Engineering Student Assistant, PSCO — Sacramento, CA — 2015-Present

Test products for specification compliance, solder components on circuit boards for assembly of small batches of products, tune and test antennas for mobile radios, source replacements for obsolete parts.

Telecommunications Technician, PSCO — Sacramento, CA — 2012-2015

Repaired damage microwave radios, performed troubleshooting on broken multi-hop radio paths, configured new radios and installed them in radio vaults for use.

I.T. Intern, Department of General Services — Sacramento, CA — 2008-2009

Helped maintain computer hardware and software for the city of Sacramento.

SKILLS

MATLAB, LabVIEW, EAGLE PCB design, AutoCAD including 3-D modeling, C language, SPICE, HTML, PHP, SQL, Mach3 CNC, MPLAB IDE, PSCAD, Advanced Design System, advanced training in full Microsoft Office suite

PROJECTS

Robotic Manipulator Controller, Electric Mountainboard with Wireless Controller, Custom Quadcopter, DC to DC Portable Power Converter, Irrigation Control System

Jamesson A. Kaupanger

OBJECTIVE:

Secure a career in the field of electronics engineering.

EDUCATION:

Bachelor of Science, Electrical and Electronic Engineering, CSU Sacramento. Overall GPA: 3.269, Graduating Spring 2017

RELATED COURSES:

- Analog & Mixed Signal IC Design
- Advanced Analog Integrated Circuits Lab
- CMOS & VLSI
- Advanced Verilog *
- Electronics II
- Electronics I
- Intro to Logic Design
- Physical Electronics
- Computational Methods (C & MATLAB)
- Network Analysis
- Intro to Microprocessors
- Intro to Feedback Systems
- Signals and Systems

** In progress as of Spring 2017*

KNOWLEDGE AND SKILLS

Programming:
Verilog, C, MATLAB

Software Applications:
Cadence Virtuoso, Eagle PCB, OrCAD Capture CIS, Xilinx ISE, MATLAB, AutoCAD, Visual Studio, Microsoft Office

Tools:
Oscilloscope, Multimeter, Signal Generator, Raspberry Pi, Analog Discovery 2, Parallax Propeller, Intel Edison

Design:
Analog Circuit Design, Digital Circuit Design, Circuit Layout

PROJECT EXPERIENCE:

Robot Arm Precision Controller - Senior Design Group Project

Designed a system to control a robotic arm with motions of the user's arm and hand. Three inertial measurement units, two along the arm and one on the back of the hand, determine a point and orientation in 3D space. The program performs forward kinematic equations to convert point and orientation to joint variables for the arm. Secondary module allows arm to be locked in position, as well as control sensitivity of movement. Design implemented using Intel Edison microcontroller.

Light-Sensing Alarm Clock

A Propeller microcontroller was used to take data from three photoresistors and average them together to get an ambient light level of a room. Using a predetermined "brightness level", the Propeller would serially communicate to an Arduino Max32 whether the room brightness level indicated "day" or "night". User could input time and set the alarm through a 3x4 keypad. The alarm would sound if the alarm time was reached or if the room got bright enough.

WORK EXPERIENCE:

Microelectronics Engineer - Intern | **Defense Microelectronics Activity** | 10/16 to present

Integrated circuit sample preparation for failure analysis and forensic engineering.

Electrical Engineering Intern | **M. Neils Engineering** | 8/15- 10/16

AutoCAD drafting, determining building compliance with Title 24, maintaining as-builts, document archiving, and completing panel schedules.

Education Representative | **Tim's Music** | 5/08- 8/15

Regularly traveled a route to visit far-away schools, picked up instruments in need of repair, dropped off repaired instruments, and delivered goods and supplies the teacher had ordered. Additionally: customer service representative, sales associate, inventory specialist, and assistant repair technician.

ACTIVITIES AND ACCOMPLISHMENTS:

Eagle Scout
Full-time missionary for two years